The Knowing Camera 2: Recognizing and Annotating Places-of-Interest in Smartphone Photos

Pai Peng [†] Lidan Shou [†♮] Ke Chen [†] Gang Chen [†] Sai Wu[†]
[♮] State Key Lab of CAD&CG
[†] College of Computer Science and Technology
Zhejiang University
Hangzhou, China
{pengpai_sh, should, chenk, cg, wusai}@zju.edu.cn

ABSTRACT

This paper presents a project called Knowing Camera for real-time recognizing and annotating places-of-interest(POI) in smartphone photos, with the availability of online geotagged images of such places. We propose a "Spatial+Visual" (S+V) framework which consists of a probabilistic field-of-view model in the spatial phase and sparse coding similarity metric in the visual phase to recognize phone-captured POIs. Moreover, we put forward an offline Collaborative Salient Area (COSTAR) mining algorithm to detect common visual features (called Costars) among the noisy photos geotagged on each POI, thus to clean the geotagged image database. The mining result can be utilized to annotate the region-of-interest on the query image during the online query processing. Besides, this mining procedure further improves the efficiency and accuracy of the S+V framework. Our experiments in the real-world and Oxford 5K datasets show promising recognition and annotation performances of the proposed approach, and that the proposed COSTAR mining technique outperforms state-of-the-art approach.

Categories and Subject Descriptors

H.3.3 [**Information Storage and Retrieval**]: Information Search and Retrieval

Keywords

location-based service; places-of-interest; image recognition

1. INTRODUCTION

Nowadays, with the wide proliferation of smartphones, the world is being captured through millions of phone cameras, and then displayed in images via Internet social applications (e.g. Facebook and Flickr) to an enormous audience. These images, ever increasing in numbers at an unprecedented rate, comprise a rich and useful cyber data-source for the physical world.

Think of a tourist visiting an unfamiliar city holding a GPS-equipped smartphone. Some of the greatest conveniences she could have are: (1) Upon taking a photo of a place (e.g. a tower or a coffee shop) using her mobile phone, the gadget displays in seconds a pop-up, showing the name of the place, probably with a URL directing to further information about it; (2) Afterwards, her friends would be able to watch her online photos, automatically annotated with respective place names, without requesting for time-consuming and error-prone tagging. This paper reports our recent work on the *Knowing Camera* (KC) project, which aims at developing a system for recognizing and annotating outdoor Places-of-Interest (POIs) captured in smartphone photos, relying on geotagged photo sharing Web services (for example Flickr).

The main requirements for KC are two-fold: (1) First, it has to recognize the POI being captured in the smartphone photo; (2) Second, it needs to annotate the respective screen region in the photo containing the POI. We address the first problem by presenting a general "spatial + visual" (S+V) framework. Then we extend our work to solve the second problem with a COSTAR mining algorithm. Notably, the output of the mining algorithm can be used to further enhance our solution to the first problem, namely POI recognition.

Motivating The S+V Framework

We shall start by introducing the motivation of our solution to POI recognition. Generally, there are two categories of existing techniques towards POI recognition, namely the *spatial* techniques and the *visual* ones.

- A naive spatial approach uses location-based method to find POIs in the vicinity of the query's GPS location. Unfortunately, this approach is flawed as the place being captured could be one mile away, not to mention dozens of other POIs which are closer to the camera. To make things even worse, the GPS location received by phone is known to be erroneous. Although the number of candidate POIs can be further reduced using the camera geometries (known as the *field-of-view* or FOV), those in the same FOV are still hard to discriminate. Moreover, the camera geometries read from phone sensors are also considerably erroneous.

- In contrast, a typical visual approach computes the *visual similarity* between the query image and online geo-tagged photos, which are widely available for many POIs on the global map (e.g. Flickr and Panaramio). By visually matching the online photos which are associated with the nearby POIs, a ranking of these POIs can be obtained. The problem with the visual approach is the large and impure image database to be searched on. On one hand, it is just too expensive to plough through all the images with similar visual

SIGIR'14, July 6–11, 2014, Gold Coast, Queensland, Australia.

http://dx.doi.org/10.1145/2600428.2609557.

features. On the other hand, the search effectiveness is often harmed by the noises contained in the database.

To address the problems with either of the above two approaches, we employ a general S+V framework which combines the spatial and visual techniques. Our framework relies on the following observations: (1) Though it is easy to acquire the camera geometries (FOV) of photos shot from a smartphone, it is not the case for those from the online image sharing sites. This implies that images in the database are not accompanied by their respective FOV information. (2) The sensor readings in cameras are erroneous. Thus, the parameters for phone direction, viewing angles, and camera location may all contain uncertainty. (3) As online photos associated with each POI contain high impurity, simply computing the visual similarity between the query and POI-tagged photos leads to indiscriminating results – similarity values which are indistinct among several POIs. Moreover, it must be noted that performing visual clustering does not help in such case.

In view of the above observations, we exploit the synergy of a *probabilistic FOV model* and a *sparse-coding visual matching* technique to facilitate POI recognition. The former gives each POI around the photographer a *likelihood* of being captured by the camera while the latter makes the resultant visual similarity reasonably discriminative. It must be noted, however, that although our S+V framework performs well in recognizing the POI of a query photo, it cannot remove the noises in the database. Nor does it annotate the query photo, which itself may contain noises.

Motivating Collaborative Salient Area Mining

The need for annotating the screen region of a POI in a photo is obvious – people want to know about and highlight places-of-interest. However, automatic object annotation is known to be a challenging task, as knowledge (or semantics) of the objects in images cannot be easily acquired from the visual features.

We propose to achieve automatic region annotation by making use of the crowd intelligence in the geotagged photos shared on the Web. Our approach is motivated by two observations:

(1) As mentioned above, the social geotagged photos of POIs in the database usually contain not only rich visual features matching those in the query photo, but also visual and semantic noises. In fact, many of the POIs are dominated by semantically irrelevant images. It makes little sense to compute visual similarity on noises. A simple study of numerous photos associated with POIs could reveal that, the "truly" visually similar ones (judged by human perception) comprise only a small portion of all. The issue is clearly demonstrated in Figure 1, which contains the associated photos of a few POIs. These noisy photos can be classified into two types: the *irrelevant* photos which do not show any outdoor appearance of the POIs, and the *noisy* ones which include not only some visual data of the POIs but also some other objects, such as human beings, plants, and vehicles etc. Therefore, the key issue for region annotation is to identify the "useful visual information" in the database while discarding the noises.

(2) Among the POI-bound photos in the database, many of the noisy ones share common visual features in groups, as the photographers have their crowd knowledge of the POIs included in the photos. This motivates us to treat similar visual feature points as *items*, and then conduct an offline frequent itemset mining algorithm to find the common visual feature points, which appear in multiple photos in the database. This mining process is called COSTAR (COllaborative SalienT ARea) mining, and literally each of these frequent items is named a *Costar*. Subsequently, the run-time visual matching (during query processing) can be done merely on the Costars, without involving the great number of noisy features. As a result, both the effectiveness and the efficiency of the visual matching can be improved.

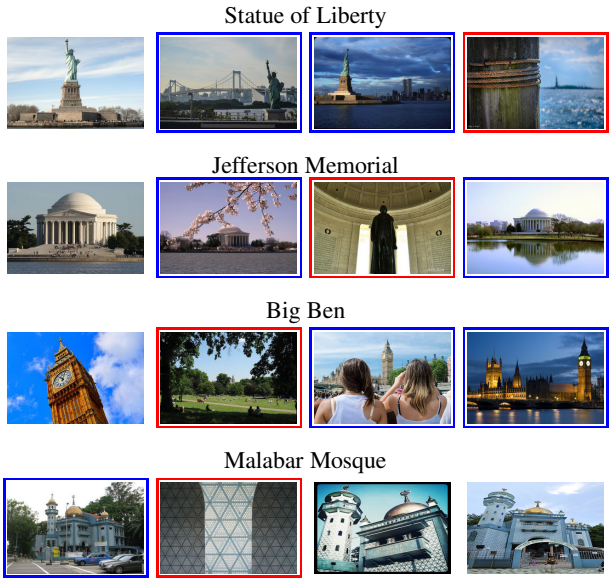


Figure 1: Flickr photos tagged by their POI names. The red-framed ones indicate the *irrelevant* photos which do not show the outdoor appearances of the respective places while the blue-framed ones indicate the *noisy* ones with occlusions such as people, vehicles and trees.

By mapping the Costars (feature points) in the screen space, all the contributing noisy photos can have their salient regions marked in offline processing. Photos without Costars are considered irrelevant and can be expunged from the database. It is also possible to mark the Costars on the query image in real-time during the online query processing. Such marking provides good basis for on-screen object annotation.

Contributions

Our main contributions are summarized as follows:

- We present the Spatial+Visual framework for accessing a set of POIs and their associated online photos, thus to support recognition of POIs from smartphone pictures (Section 3). The framework can be easily expanded to Web-scale.
- We describe the detailed techniques employed in the S+V framework, including the Probabilistic FOV (pFOV) model and the Sparse Coding Similarity metric (in Section 4).
- Based on the S+V framework, we present a COSTAR Mining algorithm to detect the salient features appearing in both the relevant images of the database and the query (Section 5). We also describe the technique to expedite the mining process.
- Our experiments (Section 6) on a real phone-captured dataset and the well-known Oxford dataset show the effectiveness of the proposed approach in both POI recognition and salient region detection.

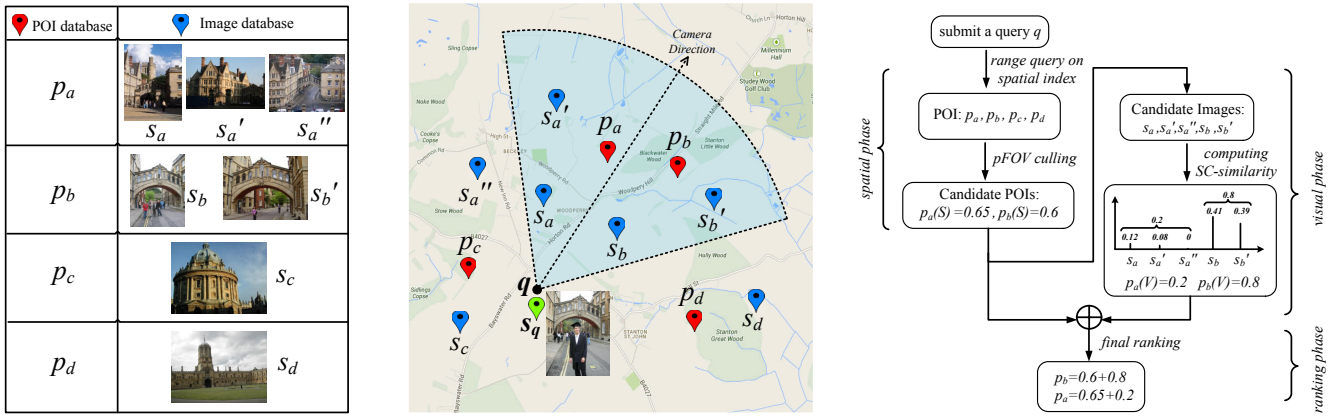


Figure 2: Basic S+V recognition process on a toy dataset. When a query is issued at q, a set of photos (blue markers) are selected. Their respective POIs (red markers) are filtered by pFOV culling. POI p_a and p_b remain as the *candidate POIs*, and their associated images become the *candidate images*. Each candidate POI is given a probabilistic geo-relevance with regard to the query. Next, the *SC-similarity* discriminates these POIs in the visual phase. Combining the spatial and visual relevances gives the final ranking.

2. RELATED WORK

Image retrieval and recognition There exist numerous research works trying to recognize or retrieve images by visual similarity. Many of them are based on the bag-of-visual-words model[18, 16, 15] which is inspired from information retrieval and text mining. The approaches vary widely from query expansion[7, 6] to soft assignment[17, 10], all aiming at increasing recall while maintaining high precision. However, our goal is not to retrieve all similar images given a query image, but to recognize the captured object. Therefore the above work cannot readily satisfy our need for specific POI recognition. The work in [16] attempts to recognize campus buildings by ranking photos in descending order of their visual similarity to a query. However, the method is hard to scale out to a massive dataset as it utilizes pure visual techniques.

A few related work attempt to recognize the place name given a geo-tagged image[22, 12, 4, 5], with the objective similar to ours. The work in [22] performs landmark mining and recognition relying on two information sources, namely GPS-tagged photos from photo-sharing website and travel guide articles. The method constructs an undirected weighted match region graph via a learning process on the landmark images, and then utilizes the graph for recognition. Another landmark recognition work[12] attempts to reconstruct the 3D structure modeled from the landmark image collection using a so-called iconic scene graph. The main drawback of these works is that they do not work for non-landmark places, which are possibly tagged by only a handful of images. The work in [5] is able to achieve city-scale landmark identification on mobile devices. However, the scheme requires the visual database to be constructed by a mobile vehicle which collects the omnidirectional street-level image data of a city. Although the method delivers attractive output, it is impractical to be deployed at a global scale due to its high cost. Moreover, the method cannot work for POIs located off the street.

The most relevant work is [4], which also combines the spatial and visual approaches in a layered structure. On the first layer, spatial partitioning is employed to retrieve a set of candidate images which are organized in visual clusters during precomputation. On the second layer, visual similarity is compared against each visual cluster to retrieve the visually similar photos. Unfortunately, this method cannot address our problem, because (i) a photo visually similar to a query may not be properly tagged by a POI, and (i-i) even if each photo is tagged correctly, visually similar photos might be tagged by different POIs. Thus it is still impossible to recognize the ONE place that is queried for. Another problem with the method is that it requires very expensive precomputation (visual clustering of images). As the database scales out, the cost to maintain the visual data structure is also expensive.

Salient region mining and detection A lot of work exist in the field of *salient region detection* which aims at detecting visually salient image regions[3, 21, 13]. These works typically produce a saliency map as the output, so that the region with high intensity in the map is expected to contain the object of interest. Unfortunately, these algorithms always perform poorly when dealing with complicated real-world scenes. When processing photos containing humans, vehicles, and other objects, the conventional saliency detection algorithm can hardly produce correct output for places-of-interest, which are typically in the background of the photos.

We consider a salient region as a set of local features which co-occur in a number of images of the POI. This idea is inspired by the work in [19], which is actually not aimed at saliency detection. In [19], a small subset of local features are selected as "useful features" for recognition if two photos in the dataset both contain them. This method can filtered out noises in the photos, such as people or vehicles, to some limited extent. However, the filtering is too relaxed as any object appearing for twice in the dataset would be taken as 'useful'. Thus, the method proposed in [19] cannot be effective for saliency detection. Another problem with the work is that it adopts the very expensive RANSAC technique to detect false-positive matched features before the useful-feature extraction. Thus, it cannot afford either efficiency or robustness when producing the final output.

In our solution, we attempt to extract the salient region from a more strict filtering mechanism. As a result, the output from our algorithm would be more meaningful and resilient to noises, as is shown in our experiments.

3. OVERVIEW OF KNOWING CAMERA

KC uses a Spatial+Visual framework for POI recognition. The data in our framework consists of (1) *a POI database* denoted by P, where each point $p \in P$ has a geo-location $p.loc$; and (2) *an image database* denoted by S, where each photo $s \in S$ has a geotag

location $s.loc$ and is also associated with one POI in P. To expedite spatial access, we also employ an *R-tree* indexing all images in S by their locations $s.loc$.

Generally, a recognition query q contains a query image $q.img$, and its camera geometries stored in $q.fov$, including its GPS location, the angles of view, the maximum visible distance, and the direction of the camera. Each query is processed in three consecutive phases, namely the *spatial phase*, the *visual phase*, and the final *ranking phase*, as illustrated in Figure 2.

Furthermore, we enhance the above framework with a novel offline process called COSTAR mining. This technique can benefit our framework in two ways: First, its output, which we call *Costars*, can be used to improve the efficiency of the aforementioned visual phase. Second, the pre-computed Costars facilitate real-time annotation of the query photo.

For better readability, we will present an overview of all these techniques in two separate parts, namely the basic framework, and the COSTAR enhancements.

3.1 The S+V Framework

We describe the basic S+V Framework in this section.

3.1.1 The Spatial Phase

The spatial phase takes the camera location as the center and makes a 2D square range query on the R-tree to retrieve images which are geotagged in the query box. These images are called "local images".

Next, the POIs associated with the local images undergo a *probabilistic FOV culling* procedure which removes the POIs that are geometrically impossible to appear in the query image. Images associated with any culled POI are deleted from the local image set. The remaining local images now become the *candidate images*, and their associated POIs are considered the *candidate POIs*. One important issue to note is that, unlike a candidate image, a candidate POI does not have to be inside the query box, e.g. point p_e in Figure 4.

Subsequently, we compute for each candidate POI p a geometric-relevance value (denoted by *geo-relevance*), which indicates the geographical probability that p appears in the query pFOV.

3.1.2 The Visual Phase

Given a list of candidate images, we consider each of their respective visual feature vectors as a basis signal, and employ the *sparse coding* technique to obtain a linear combination of the basis vectors, which maximally reproduces the original signal (the feature vector of the original image). We shall justify the reason for using sparse coding later.

Each weight in the linear combination indicates the contribution of the respective basis signal to reproduce the query image. These weight values are then used as the visual similarity for each respective candidate image with regard to the query. Finally, these weight values are aggregated by different POIs, so that each POI receives the sum of all the weights from its associated candidate images. The aggregated weights are also called *sparse coding similarity* (SC-similarity) for each POI.

3.1.3 The Ranking Phase

We use a straightforward method to combine the spatial and visual phase in a weighted voting:

$$score = \lambda \cdot geo\text{-}relevance + (1 - \lambda) \cdot SC\text{-}similarity, \quad (1)$$

where $0 \leq \lambda \leq 1$ controls the preference for the spatial model or the visual one. Then the POI with the top ranking score is the final recognition result.

3.2 COSTAR Enhancements

A Costar in a photo is defined as a robust and useful feature point located in the outdoor appearance of the captured POI. There might be hundreds of Costars in a photo. The salient region of a photo is a screen region covering all its Costars. We shall first look at the offline mining process of finding Costars for images in the database, and then describe the Costar-augmented visual matching and annotation techniques.

3.2.1 Offline Mining

We use a certain category of visual features called *local features* [20] for COSTAR mining. Each image q in the database is considered a fake "query" and undergoes a visual reconstruction process, where the query feature vector is attempted to be reproduced by neighboring images (those geotagged in its vicinity). In many cases, this will give a set of feature points that are frequently matched in the neighboring images. These feature points are taken as the Costars of q. However, if the reconstruction fails, or that q cannot be reproduced from neighboring images within an acceptable error, then q is considered an irrelevant image and can be expunged from the database. As a result, the database will be cleaned and contain only relevant (and probably noisy) images with their respective Costars marked.

3.2.2 The Visual Phase With Costars

With the Costars detected for all images in the database, we can improve the visual phase of the S+V framework. As all irrelevant images are discarded from the database, the population of the candidate images to be considered in the visual phase is significantly reduced. Then we conduct the sparse coding technique on the remaining photos (excluding those discarded in offline mining), and take the aggregated weight values as the sparse coding similarity for each POI. Since the number of candidates becomes smaller, the visual matching is expected to be more efficient.

3.2.3 Online Query Annotation

Now the online salient region annotation for the query photo is straightforward. We only need to find the Costars in the query image which also appear in the visually similar images, or those with high SC-similarity values. These Costars can be marked on the query image, and their minimum bounding rectangle can be plotted to annotate the salient region of the POI. As the number of visually similar images contributing Costars is small, the annotation can be done in real-time.

4. SPATIAL AND VISUAL TECHNIQUES

In this section, we will give the details of computation in the spatial phase and visual one.

4.1 Spatial Technique

4.1.1 The Probabilistic FOV Model

Given a query q, figure 3 illustrates the probabilistic FOV (p-FOV) model, which is derived from the conventional FOV (field-of-view) model. The conventional FOV model consists of 4 parameters, namely the camera location q, the camera direction $\vec{v}$, the viewing angle α, and the maximum visible distance R. However, the first three quantities contain an abundance of uncertainty due to device errors and such uncertainty of FOV parameters define the

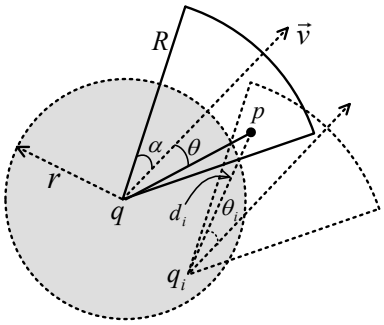


Figure 3: The probabilistic FOV model consisting of 4 parameters, namely the camera location q, the camera direction $\vec{v}$, the camera viewing angle α, and the maximum visible distance R.

probabilistic FOV of the camera. Specifically, we model the probability distribution function of POI p being captured by an exact FOV at q as the following function:

$$P(\theta, d) = e^{-\frac{\|\theta\|^2}{2{\sigma_1}^2}} \cdot e^{-\frac{\|d\|^2}{2{\sigma_2}^2}} \tag{2}$$

where θ is the angle between $\vec{v}$ and $\overrightarrow{qp}$ and $d = \|pq\|$. σ_1 and σ_2 are parameters which ensure that $P(\theta, d)$ is a negligible value (≈ 0) if p is outside the FOV region.

Consider the probabilistic FOV model where the first three parameters become uncertain, the probability of a POI p captured by a query image is a cumulative distribution function which is given by an integral of Equation 2

$$geo\text{-}relevance = \int_Q e^{-\frac{\|\theta\|^2}{2{\sigma_1}^2}} \cdot e^{-\frac{\|d\|^2}{2{\sigma_2}^2}} \mathrm{d}q \tag{3}$$

where Q is the circular region for the Gaussian distribution of the camera location with radius r.

Evaluating Equation 3 is difficult. Alternatively, we use a sampling based method to solve it approximately. We divide the integral area(the shaded circle in Figure 3) into equal-sized small grid cells so that $P(\theta, d)$ is considered to be identical for all q_i points in the same cell Δq. Thus the cumulative probability for Δq is $P(\theta, d) \cdot \Delta q$. The marginal cells which overlap the circle boundary are ignored as their cumulative probabilities are very small. As a result, the whole integral can be approximated by

$$geo\text{-}relevance \approx \sum_{i=1}^{N} e^{-\frac{\|\theta_i\|^2}{2{\sigma_1}^2}} \cdot e^{-\frac{\|d_i\|^2}{2{\sigma_2}^2}} \cdot \Delta q \tag{4}$$

where N is the number of cells.

4.1.2 Probabilistic FOV Culling

It can be seen that for any POI p appearing in the query FOV, there must exist a probabilistic FOV instance (at q_i) which contains p. In other words, the union of all pFOV instances must cover all possible candidate POIs. This observation leads to a *pFOV culling* algorithm, which relies on the following Lemma.

Lemma 1: *All candidate POIs whose locations are outside the union of all pFOV instances can be discarded.*

A sample process of pFOV culling is given in Figure 4, where POIs located in the shaded region can be culled. Thus, POIs p_c, p_d can be culled, because their actual FOVs cannot be any pFOV instance for q. However, p_e cannot be culled because it is within the reach of a possible pFOV instance (q'), even when it is outside the query box.

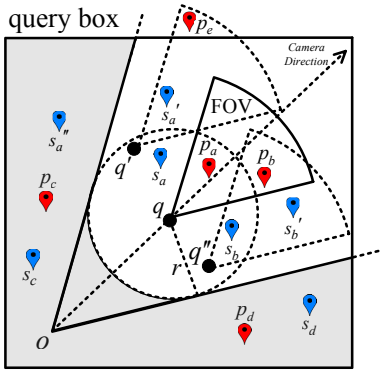


Figure 4: Illustration of pFOV culling and candidate images/POIs. The blue markers indicate the geotagged locations of the images $\{s_i\}$, while the red ones $\{p_i\}$ indicate their respective POIs. The circular sectors centered at q, q', and q'' illustrate the pFOV instances given a camera GPS reading at location q.

4.2 Visual Technique

To compute the visual similarity, each candidate image is represented as a bag-of-visual-words column vector. Let D be the matrix where each column is the vector for each candidate image in C. Then the problem can be described as: Given a query image $\mathbf{x}$, can we represent it as a linear combination of other candidate images(columns in D). This is a typical problem of sparse coding and the objection function is given by

$$\min_{\omega} \frac{1}{2} \|D\mathbf{w} - \mathbf{x}\|^2 + \alpha \|\mathbf{w}\|_1 \tag{5}$$

where α controls the sparsity in $\mathbf{w}$ and $\|\mathbf{w}\|_1$ is the l_1-norm of the parameter vector. Furthermore, the original query image $\mathbf{x}$ can be reconstructed by multiplying D by the sparse code $\mathbf{w}$.

This optimization problem is called Lasso and can be solved by the least angle regression(LAR) approach [8]. The optimization result is a sparse code as $\mathbf{w} = (\omega_0, \omega_1, ..., \omega_p)^T$, where ω_i indicates the contribution of the ith candidate image to reproduce the query image. Weight value ω_i is then used as the *visual similarity* between candidate image i and the query. Finally, these weight values are aggregated by different POIs, so that each POI receives the sum of all the weights from its associated candidate images. The aggregated weights are called *sparse coding similarity* (SC-similarity) for each POI.

$$SC\text{-}similarity = \frac{\sum_{d_i \in POI} \omega_i}{\sum \omega_i} \tag{6}$$

5. COSTAR MINING

In this section, we describe the techniques of mining and using Costars in our framework. We first present the *offline Costar mining algorithm*. Then we propose two additional techniques to address issues with the mining algorithm. One is a post-procedure called *grid-based denoising*, which removes non-robust outliers produced by the mining algorithm. The other is an approximate nearest neighbor technique to expedite the mining process. The entire process of Costar mining is depicted in Figure 5. In the end, we briefly describe the *online Costar annotation* procedure.

Figure 5: COSTAR mining example. Note that the mining process may lead false-positive outliers which will be eliminated by grid based denoising.

5.1 Definition of Costar

Definition 1: Given a set of photos $S = \{p_i\}$, each photo p_i is associated with a GPS location $p_i.loc$ and a set of detected local features (e.g. SIFT), denoted by $F(p_i) = \{f_j\}$ $(j = i_1, i_2, \ldots)$. A local feature f_m of p_i is called a *Costar*, if f_m also appears in at least c photos in S other than p_i. Formally, if $\exists f_m \in F(p_i)$ so that set $D = \{p_k | f_m \in F(p_k), \forall p_k \in S \wedge k \neq i\}$ contains at least c photos, or $|D| \geq c$, we say f_m is a Costar. The set of all Costars of photo p_i is denoted by $CST(p_i)$.

Note that a photo may contain none or numerous Costars, depending on the quality of the photo itself and its context (the photo set S). Given an image dataset S, a photo in it may contain no Costars if it is visually irrelevant to any other photos in S. On the contrary, it may contain thousands of them, if it finds many similar local features among the other images in S.

5.2 Basic Offline Mining Algorithm

Our target of COSTAR mining is to find the Costars for each photo in the database. However, it is unnecessary to search for Costars across the whole database. The reason is that, given an image q at a geolocation $q.loc$, it makes little sense to search for meaningful Costars among photos which are located at a great distance from $q.loc$, as those very distant photos cannot be capturing the same POI in q. Thus, we can reduce the search space of mining to a limited geographical area using the spatial index.

The basic mining algorithm is illustrated in Algorithm 1. We start from a noisy photo database S in which each photo is associated with a geotag and a set of local features. We use SIFT[14] points as the local features in our implementation due to their robustness. Firstly, for each photo x_i in the database, we perform a range query on a spatial index (R-tree in our implementation), which produces a set of nearby photos named D. Secondly, a visual reconstruction process is conducted which attempts to reproduce x_i with D via sparse coding technique. If the reconstruction attempt fails, x_i is considered as an irrelevant photo and it can be discarded from the database. Next, we select the set of images which have high SC-similarities with x_i and denote it by D'. For each photo d in D', we perform a SIFT feature matching between x_i and d to obtain a set of matched features (items) in x_i. When we finish matching all photos in D' to x_i, those frequently matched items in x_i are taken as the Costars.

Note that our mining algorithm is not restricted to any particular visual similarity metric. One can easily replace the SC-similarity on Line 8 with other metrics. However, the SC-similarity proved to perform very well in selecting the truly relevant images. In our implementation, the $Feature\_Match$ operator on Line 12 uses the classic distance ratio from [14].

Algorithm 1: COSTAR mining algorithm

Input: a photo database $S = \{x_i\}$, each photo x_i is attached a geotag and a set of local features $F_i = \{f_k\}$.
Result: discard irrelevant photos from S and detect Costars for the rest photos.

```
1  for x_i ∈ S do
       /* D is a set of nearby photos                    */
2      D = {d_j} ⟵ range query on R-tree index;
3      w ⟵ compute sparse code with D in Eq.5;
4      if ||Dw − x_i||^2 > ε then
5          x_i is an irrelevant photo;
6          discard x_i from S;
7      else
           /* x_i can be reconstructed by D
              within an acceptable error                 */
8          Sim ⟵ compute SC-similarity in Eq.6;
           /* D' is a set of similar images              */
9          D' = {d_j | Sim_j > τ, d_j ∈ D};
10         c[ ] ⟵ 0 ;    /* counting the matches        */
11         for d_j ∈ D' do
12             {f_t} ⟵ Feature_Match(x_i, d_j);
13             for f ∈ {f_t} do
14                 c[t] ⟵ c[t] + 1;
15         Costars_{x_i} = {f_k | c[k] > c, f_k ∈ F_i};
```

There are three parameters in the algorithm, namely ε, τ, and c, which control the bounds for reconstruction error, the SC-similarity, and the number of Costars respectively. Varying these parameters impact the COSTAR mining results. We will discuss the tuning of these parameters in the experiment section.

5.3 Grid based denoising

As indicated in the fourth column of Figure 5, the Costars produced by the basic mining algorithm may contain a few false-positive outliers, which are isolated from the others. These outliers are often far away from the true-positives on the screen, having some different colors or textures with them. Therefore, a post-processing procedure is needed to eliminate the outliers in the output of the basic mining algorithm.

To remove outlier Costars in an image, we split the image in screen space into a number of equal-sized grid cells, eg. 8×8 cells. For each cell , we need to evaluate (1) the number of Costars contained in it – a cell containing very few Costars indicates the

existence of outliers; and (2) the difference of the current cell with its surrounding cells in low-level visual features – great difference in colors or textures also indicate outliers.

Thus we define for each cell g_k a *grid saliency value* as

$$s(g_k) = w(g_k) \sum_{g_i \neq g_k} S(g_i, g_k) \cdot e^{-d/\sigma_g^2} \quad (7)$$

where $w(g_k)$ is the number of Costars contained in cell g_k, $S(\cdot,\cdot)$ is the low-level visual feature similarity between cell g_k and another cell in the same image. e^{-d/σ_g^2} is a penalizing factor which reduces when the distance d between cell g_k and g_i increases, indicating that the impact of g_i diminishes by its distance to g_k. A cell with $s(g_k)$ value less than a threshold is considered to be an outlier region, and the Costars contained in it would be eliminated. In our prototype, $S(\cdot,\cdot)$ is computed as the naive color histogram similarity.

5.4 Speedup with ANN

The $Feature\_Match$ operation in Algorithm 1 is very costly, as it adopts the metric evaluation proposed in [14]. Specifically, the metric evaluates the ratio between the nearest and the second nearest neighbor by the Euclidean distance. Matches that are greater than a threshold value are rejected as false matches. Thus, the high matching cost is largely due to the operation of computing the nearest and the second nearest neighbors in the feature space. As reported in [11], it takes $2.15378 * 10^4$ seconds to compare a pair of two 300×240 images finding 271 matches, not to mention the more expensive SIFT matching between high quality photos.

To speed up this matching process, we adopt the Approximate Nearest Neighbor (ANN) technique, which retrieves the top-k approximate nearest-neighbours at significantly reduced cost compared to exact NN search in high dimensions. In our method, we conduct ANN to construct an index for the target image, so the subsequent nearest neighbor queries can substantially benefit from it [1]. The degree of approximation and the query performance is balanced by a parameter ρ (which was originally denoted by ϵ in [1]). Specifically, a ρ value of 0 degenerates the scheme to exact NN search and a greater ρ value leads to higher efficiency at the expense of accuracy. The effect of tuning ρ will be demonstrated in the experiment.

5.5 Online Annotation

Online annotation is similar to the offline mining process. Given a query photo q, we utilize the S+V framework to find the top ranking POI p. As a byproduct, we also obtain a set of visually similar photos during the visual phase of the recognition. This set of photos, denoted by D', then undergo the feature matching procedure as described between Line 11-15 in Algorithm 1. More importantly, q is only matched with the precomputed Costar features of the images in D'. The cost of such matching is much less than that on the original features. With the additional help of ANN, the annotation can be done in milliseconds.

6. EXPERIMENTAL RESULTS

In this section, we evaluate the performance of our proposed framework on two real-world datasets, *Campus 13K* which is collected by the students via our Android application and *Oxford Buildings 5K dataset*[2], a well-known published dataset as a benchmark for image recognition and annotation. Note that images in the Oxford dataset do not contain GPS information. Therefore, it is only used in the annotation experiment to compare with state-of-the-art method.

6.1 Experiment Setting

6.1.1 Dataset Description

We extract SIFT features as our local features for each photo in both datasets. Table 1 lists the details.

Table 1: Dataset Details

dataset	#images	image size	#features/image
Campus13K	13236	640×480	1014
Oxford5K	5061	1024×768	3830

Campus 13K + Flickr 13K Our S+V approach requires the FOV information to compute the probabilities of POIs. However, most images from photo sharing Web Service (e.g., Flickr) do not include such information. To address the issue, we develop a camera application for Android system to allow users to take photos with FOV tags. The FOV tags are automatically attached to photos, and we ask our students to collect an image dataset of the campus using the application. In particular, we employ 5 students to shoot a total number of 13236 photos for 322 POIs. Each POI is captured for many times from arbitrary angles by at least 5 students, so as to deliberately introduce noises in the photos. After shooting, the students are required to tag the POI name for each photo. These POI names are then used as our ground-truth in the experiment. The dataset contains 10 *landmark POIs*, each associated with more than 100 images; and 161 *trivial POIs*, each associated with less than 50 images. To simulate a large amount of irrelevant images in the real word, we add in each POI an equal number of randomly selected Flickr photos (Flickr13K). Thus the combined dataset (Campus13K + Flickr13K) contains 10 landmarks with more than 200 images, and 161 trivial ones having less than 100 images. It takes on average 27 (90) seconds to mine Costars for a trivial (landmark) POI, on a commodity PC with 8 cores and 16GB memory. Although the Campus13K dataset is small in size, it can effectively simulate the density of photos in urban areas. Notably, a larger dataset which covers a greater region is not expected to affect the recognition performance, as the spatial range query ranks the POIs by their localities.

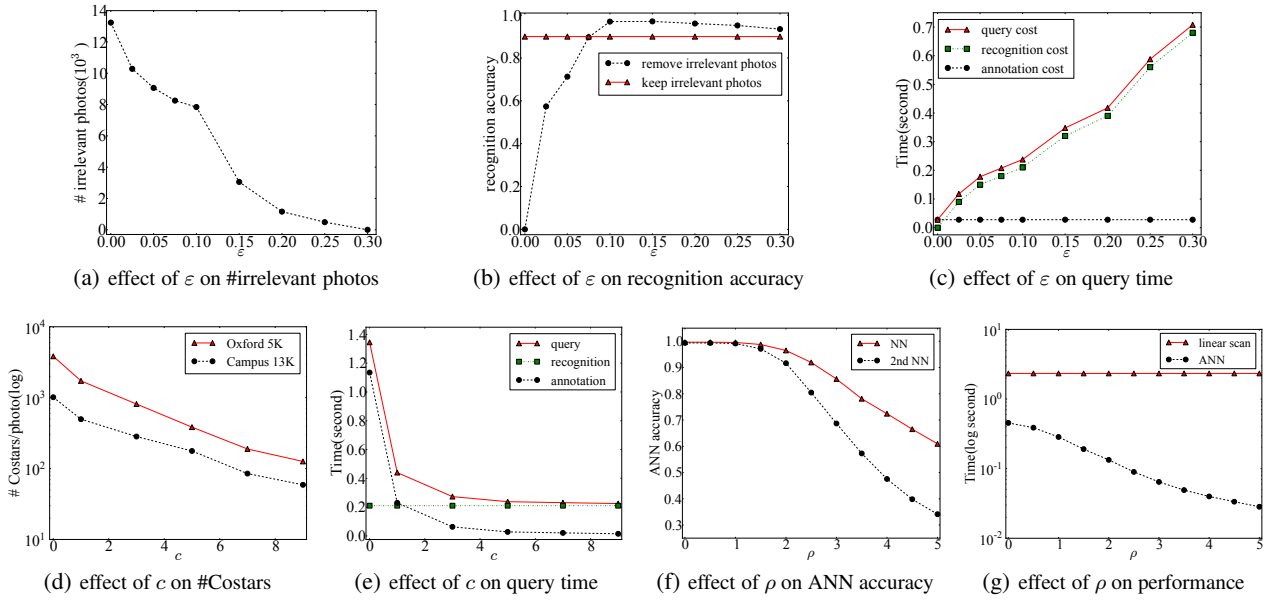
We randomly select 600 different images from the Campus13K dataset, which covers the entire POI set, as our queries. Then we submit a query with FOV information to retrieve the rest images in the dataset to evaluate the effectiveness of POI recognition.

Oxford Buildings 5K Oxford Buildings 5K is a popular dataset used in many image retrieval and recognition work. It contains 5062 high quality photos labeled with 11 different landmarks (POIs) in Oxford. Unfortunately, these photos do not contain GPS information, which means that the pFOV model cannot be applied. However, we can still detect Costars for each landmark. For the most prominent POI, the Bodleian Library, which includes 275 high quality images, it takes approximately 4 minutes to mine all its Costars. Then we compare the annotation results of our COSTAR algorithm with the algorithm proposed in [19], which selects the so-called "useful features".

6.1.2 Evaluation Metric for Recognition

Recognition Accuracy We define the recognition accuracy as the number of correctly recognized queries divided by the total number of query images.

$$Accuracy = \frac{\#correctly\ recognized\ queries}{\#total\ queries} \quad (8)$$



(a) effect of ε on #irrelevant photos (b) effect of ε on recognition accuracy (c) effect of ε on query time

(d) effect of c on #Costars (e) effect of c on query time (f) effect of ρ on ANN accuracy (g) effect of ρ on performance

Figure 6: Tuning the COSTAR mining parameters.

Baseline Approach Besides our proposed framework KC, we also implement state-of-the-art approach, bag-of-visual-words(*BOW*), as a baseline, which has a 5K-visual-vocabulary same as the one generated in [9].

6.2 Tuning Parameters for COSTAR Mining

(1)Effect of ε

ε plays a key role in judging irrelevant photos as a threshold. Apparently, smaller ε results in more irrelevant photos pruned by Algorithm 1 as shown in Figure 6(a). In the extreme case, $\varepsilon = 0$ makes all photos in the database become irrelevant since we can hardly reconstruct a query photo via a few different photos without any information loss. When ε is large, there would be fewer irrelevant photos discarded, which implies that the overhead of query processing is high. This is a trade-off between the accuracy and processing cost. In fact, the recognition accuracy does not drop sharply even for a larger ε, because our S+V framework can still filter out the irrelevant photos by pFOV culling and sparse coding techniques. The analysis above is verified by Figure 6(b) and 6(c). Note that the query cost in Figure 6(c) is actually the summation of the recognition cost and the annotation cost.

(2)Effect of c

c is an important parameter in COSTAR mining algorithm. It controls the number of detected Costars in each photo. Larger c leads to less Costars but more discrimination and robustness. In other words, we are more confident that Costars belong to the salient POI region instead of noisy ones such as trees or vehicles. Specifically, the Costars degenerate into original local features (SIFT feature in current implementation) when $c = 0$. Figure 6(d) shows the effect of varying c on both datasets, and Figure 7 illustrates a detailed example. With the number of Costars decreased, the online annotation process incurs less overhead, which is shown in Figure 6(e). We choose $c = 5$ in the following experiment for the trade off between accuracy and efficiency.

(3)Effect of ρ

We use the ANN[1] to facilitate the SIFT matching. ANN scheme needs to find the nearest neighbor and the 2nd nearest neighbor efficiently. In [1], error bound ρ controls the degree of approximation as a tuning parameter. If $\rho = 0$ (no error is allowed), exactly k(here, $k = 2$) nearest neighbors are retrieved. Obviously, larger ρ leads to a rougher estimation but lower computation cost. We illustrate the average accuracy of nearest neighbor and the 2nd nearest neighbor in the SIFT matching estimated by the ANN as well as the processing time with different error bounds in Figure 6(f) and 6(g). When $\rho < 1.5$, both NN and 2nd NN accuracy is close to 99% and the processing cost decreases dramatically compared to linear scan.

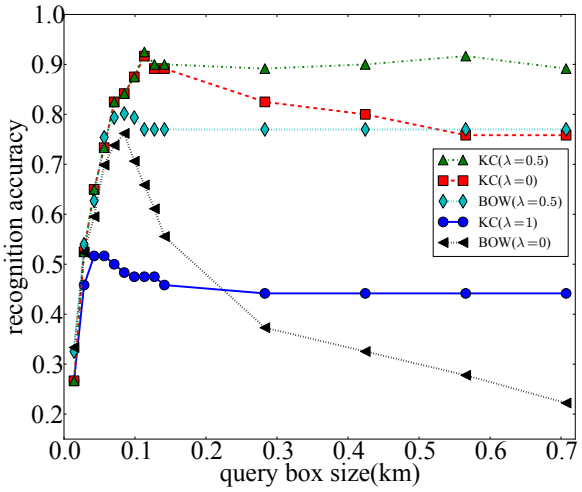
(a) c=0 (b) c=1 (c) c=3

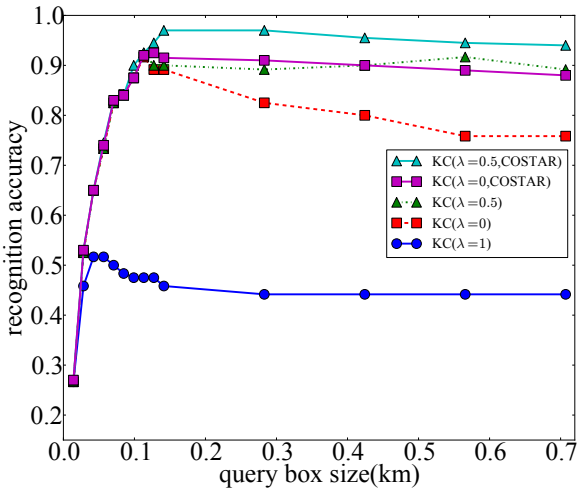
(d) c=5 (e) c=7 (f) c=9

Figure 7: An example to show Costars effected by different c.(Best viewed in color PDF)

6.3 Results of Recognition

The POI recognition accuracy of the S+V framework is illustrated in Figure 8(a), where we vary the spatial query box size and plot the results of different λ values. When $\lambda = 1$, the KC scheme de-





(b) effect of COSTAR

Figure 8: Results of Recognition Accuracy.

generates to a pure spatial method. For comparison, we also plot the accuracy of the BOW. It can be seen that (1) The pure spatial method alone does not perform well; (2) As the query box expands, the accuracy of all schemes is significantly improved (when box size $<$ 120m) because more relevant photos are added into the candidate image set. However, as the query box size increases further, the accuracy of pure SC-similarity ($\lambda = 0$) declines considerably. This is a clear indication that the recognition problem becomes more challenging as the search space increases. A more apparent performance degradation can be observed for the BOW method, due to the same reason. Fortunately, the problem in SC-similarity can be neutralized by the geo-relevance in our scheme. When $\lambda = 0.5$, the accuracy is very stable and remains to be above 90%. Such results confirm the effectiveness of our spatio-visual ranking score.

We also show the effect of COSTAR in Figure 8(b). Compared with the basic S+V framework, it can be seen that in the case of pure visual recognition ($\lambda = 0$), COSTAR mining considerably improves the accuracy as it removes irrelevant images in the database. Similar improvement can also be observed when $\lambda = 0.5$. These results verify the effectiveness of our COSTAR enhancement. Apparently, COSTAR has no impact on pure spatial approach ($\lambda = 1$). Thus we only show its result without COSTAR.

Figure 9: Sample images (left), their original features(middle), and the detected salient regions(right) in Campus13K. (Best viewed in color PDF)

6.4 Results of Annotation

Annotation Results We show a few annotation examples of Campus 13K data in Figure 9. It can be seen that the annotated regions are very well matched with the outdoor appearance of the recognized POIs.

Costars vs. "Useful Features" As mentioned before, we compare Costars with "useful features" proposed in [19] which aims at extracting useful and robust features from photos(discussed in Section 2) in Figure 10. It can be seen that our method performs better. Note that the useful features shown here differ from those illustrated in the original paper, due to the randomness introduced by the RANSAC method in [19]. This is exactly a major limitation of [19]. On the contrary, our method is much more stable and robust. Only those frequently matched features are considered useful and our proposed grid based denoising method can further eliminate the false-positive matches, thus refining our final results.

The Oxford5k dataset also contains a subset of 55 images, all manually annotated by a rectangular salient region. Taking each box region as the ground-truth, we measure the percentage of *Costars* and *Useful Features* appearing in the box (which is considered to be the positive rate). The result shows that Costars have an average positive rate of 83%, whereas the "useful features" only produce 61% positive.

7. CONCLUSION

In this paper we presented the KC project for real-time recognizing and annotating places-of-interest in smartphone photos. The proposed S+V framework aimed at the basic recognition problem, leveraging the pFOV model in the spatial phase and the SC-similarity metric in the visual phase. Based on this framework, we proposed a novel COSTAR mining algorithm to annotate salient regions in noisy photos. In addition, the output of this algorithm

Figure 10: The "useful features" in [19] (middle column) and the Costars when c=5 (right column). The original feature points are shown on the left column.(Best viewed in color PDF)

can be used to filter irrelevant images from the database. Our experiments on real-word datasets confirmed the effectiveness of the proposed techniques.

Acknowledgments

The work is supported by the National Science Foundation of China (GrantNo. 61170034) and National High Technology Research and Development Program of China (GrantNo. 2013AA040601).

8. REFERENCES

[1] http://www.cs.umd.edu/∼mount/ann/.

[2] http://www.robots.ox.ac.uk/∼vgg/data/oxbuildings/.

[3] R. Achanta, F. J. Estrada, P. Wils, and S. Sĺźsstrunk. Salient region detection and segmentation. In *ICVS*, volume 5008 of *Lecture Notes in Computer Science*, pages 66–75. Springer, 2008.

[4] Y. S. Avrithis, Y. Kalantidis, G. Tolias, and E. Spyrou. Retrieving landmark and non-landmark images from community photo collections. In *ACM Multimedia*, pages 153–162. ACM, 2010.

[5] D. M. Chen, G. Baatz, K. Koser, S. S. Tsai, R. Vedantham, T. Pylvanainen, K. Roimela, X. Chen, J. Bach, M. Pollefeys, B. Girod, and R. Grzeszczuk. City-scale landmark identification on mobile devices. In *CVPR*, pages 737–744. IEEE, 2011.

[6] O. Chum, A. Mikulĺłk, M. Perdoch, and J. Matas. Total recall ii: Query expansion revisited. In *CVPR*, pages 889–896. IEEE, 2011.

[7] O. Chum, J. Philbin, J. Sivic, M. Isard, and A. Zisserman. Total recall: Automatic query expansion with a generative feature model for object retrieval. In *ICCV*, pages 1–8. IEEE, 2007.

[8] B. Efron, T. Hastie, I. Johnstone, and R. Tibshirani. Least angle regression. *The Annals of Statistics*, 32(2):407–451, 2004.

[9] H. Jégou, M. Douze, and C. Schmid. Hamming embedding and weak geometric consistency for large scale image search. In *European Conference on Computer Vision*, volume I of *LNCS*, pages 304–317. Springer, oct 2008.

[10] H. Jegou, M. Douze, and C. Schmid. Improving bag-of-features for large scale image search. *International Journal of Computer Vision*, 87(3):316–336, 2010.

[11] L. Juan and O. Gwon. A Comparison of SIFT, PCA-SIFT and SURF. *International Journal of Image Processing (IJIP)*, 3(4):143–152.

[12] X. Li, C. Wu, C. Zach, S. Lazebnik, and J.-M. Frahm. Modeling and recognition of landmark image collections using iconic scene graphs. In *ECCV (1)*, volume 5302 of *Lecture Notes in Computer Science*, pages 427–440. Springer, 2008.

[13] T. Liu, Z. Yuan, J. Sun, J. Wang, N. Zheng, X. Tang, and H.-Y. Shum. Learning to detect a salient object. *IEEE Trans. Pattern Anal. Mach. Intell.*, 33(2):353–367, 2011.

[14] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60:91–110, 2004.

[15] D. Nistĺer and H. Stewĺęnius. Scalable recognition with a vocabulary tree. In *CVPR (2)*, pages 2161–2168. IEEE Computer Society, 2006.

[16] J. Philbin, O. Chum, M. Isard, J. Sivic, and A. Zisserman. Object retrieval with large vocabularies and fast spatial matching. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2007.

[17] J. Philbin, O. Chum, M. Isard, J. Sivic, and A. Zisserman. Lost in quantization: Improving particular object retrieval in large scale image databases. In *CVPR*. IEEE Computer Society, 2008.

[18] J. Sivic and A. Zisserman. Video google: A text retrieval approach to object matching in videos. In *ICCV*, pages 1470–1477. IEEE Computer Society, 2003.

[19] P. Turcot and D. G. Lowe. Better matching with fewer features: The selection of useful features in large database recognition problems. In *ICCV Workshop on Emergent Issues in Large Amounts of Visual Data (WS-LAVD)*, 2009.

[20] T. Tuytelaars and K. Mikolajczyk. Local invariant feature detectors: A survey. *Foundations and Trends in Computer Graphics and Vision*, 3(3):177–280, 2007.

[21] J. Wang, C. Zhang, Y. Zhou, Y. Wei, and Y. Liu. Global contrast of superpixels based salient region detection. In *CVM*, volume 7633 of *Lecture Notes in Computer Science*, pages 130–137. Springer, 2012.

[22] M. Z. Zheng, Yan-Tao, Y. Song, H. Adam, U. Buddemeier, A. Bissacco, F. Brucher, T.-S. Chua, and H. Neven. Tour the world: Building a web-scale landmark recognition engine. In *CVPR*, pages 1085–1092. IEEE, 2009.